



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,913	10/24/2003	Ray Y. Lai	5681-66303	9034
58467	7590	06/26/2008		
MHKKG/SUN				
P.O. BOX 398				
AUSTIN, TX 78767				
EXAMINER				
INGBERG, TODD D				
ART UNIT		PAPER NUMBER		
2193				
MAIL DATE		DELIVERY MODE		
06/26/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/692,913

**Applicant(s)**

LAI, RAY Y.

**Examiner**

Todd Ingberg

**Art Unit**

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 26 February 2008.  
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.  
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-17 is/are pending in the application.  
4a) Of the above claim(s) 18-92 is/are withdrawn from consideration.  
5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.  
6) ☒ Claim(s) 1-17 is/are rejected.  
7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.  
8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.  
10) ☒ The drawing(s) filed on 2/26/2008 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)  
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3) ☐ Information Disclosure Statement(s) (PTO-8508)  
Paper No(s)/Mail Date \_\_\_\_\_  
4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_  
5) ☐ Notice of Informal Patent Application  
6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

Claims 1 – 17 have been examined.

Claims 18 – 92 can be filed as Divisional Applications.

### ***Drawings***

1. New corrected drawings have been received.

### ***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)). Effective filing date of the instant application is August 18, 2003.

3. Claims 1 – 4, 8 – 10 and 13 – 14 are rejected under 35 U.S.C. 102(e) as being anticipated by “Building Web Applications with UML”, Second Edition, Jim Conallen, October 10, 1002 (Web).

### **Claim 1**

**Web** anticipates a system for integrating Web Services with a business system (Web, page 9, last paragraph), comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor (Web, page 10, Figure 2-1, and text) to:

Art Unit: 2193

generate an integrated Web Service architecture (Web, page 10, Figure 2-1 and pages 65-66 – UDDI and pages 64-65 – SOAP and pages 66 – 68, WSDL) comprising a plurality of heterogeneous components of the business system in accordance with one or more integration design patterns (Web, page 425, Master Template Pattern or classes – one example in Figure 6-11, pages 115); wherein, to generate an integrated Web Service architecture (As per above), the program instructions are further executable by the processor to: generate one or more Use Cases for the integrated Web Service (Web, page 173 – 185, 120, 216, 411—414, 101-105, 177-179, 139-141, 179-183); generate a high-level architecture for the integrated Web Service (As per above), wherein the high-level architecture identifies two or more entities of the integrated Web Service (Web, page 438) and the relationships and interactions among the entities (Web, page 177, relationship); and generate a logical architecture for the integrated Web Service according to the Use Cases (Web, pages 237-242), wherein the logical architecture identifies two or more logical components of the integrated Web Service (Web, page 237-242) and the relationship among the logical components (Web, page 237-242), and wherein the logical architecture comprises two or more layers (Web, page 237-242).

**Claim 2**

The system as recited in claim 1, wherein, to generate an integrated Web Service architecture (See the rejection for claim 1), the program instructions are further executable by the processor (See the rejection for claim 1) to: define a plurality of integration tiers (Web, pages 431 – 439, tiers), one or more basic components (Web, page 35, classes of an HTML document), and one or more Web Services technologies for integration (Web, page Explorer, page 84); and define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers (Web, pages 431 – 439, tiers).

**Claim 3**

The system as recited in claim 2, wherein the plurality of integration tiers comprises one or more of: a client tier (Web, page 431), a presentation tier (Web, page 435), a business tier (Web, page 115, Business Requirements), an integration tier (Web, page 122, Figure 6-15), and a resources tier (Web, page 438-439 or left side of 115).

**Claim 4**

The system as recited in claim 1, wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to define integration of one or more Enterprise Application Interface (EAI) products with the one or more Web Services technologies (See the rejection for claim 1 – ability to integrate more than 1 web service).

**Claim 8**

The system as recited in claim 1, wherein the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network (See the rejection for claim 1 and a web application, pages 22-23).

Art Unit: 2193

**Claim 9**

The system as recited in claim 8, wherein the integrated Web Service business system is a Business-to-Consumer system, wherein the service provider is a business service provider, and wherein the service requester is an end user (See the rejection for claim 1 and a web application, pages 22-23).

**Claim 10**

The system as recited in claim 8, wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is a business server (See the rejection for claim 1 and a web application, pages 22-23).

**Claim 13**

The system as recited in claim 1, wherein the design patterns include one or more integration design patterns See the rejection for claim 1.

**Claim 14**

The system as recited in claim 13, wherein the integration design patterns include one or more of: an Application-to-Application design pattern (See the rejection for claim 1 – basic object to object messaging built from defined classes – in figures); a Standard Build design pattern; a Hub-Spoke Replication design pattern; a Federated Replication design pattern; a Multi-Step Application integration design pattern; and a Data Exchange design pattern.

***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 5 – 7 and 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over by “Building Web Applications with UML”, Second Edition, Jim Conallen, October 10, 1002 (Web) in view of Object-Oriented Information Systems Planning and Implementation, David A. Taylor, April 10, 1992.

**Motivation to Combine Web and Taylor**

Web teaches an integrated environment to generate a web service and the ability to build classes/ patterns to support the development. It is Taylor who teaches the old and well known technique of wrapping legacy systems (Taylor, pages 296 – 299, wrappers). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine Web with Taylor, because the ability for newly developed systems to interact with existing systems reduces development cost.

**Claim 5**

The system as recited in claim 1, wherein the business system is an Enterprise business system (Taylor, pages 296 – 299, wrappers).

**Claim 6**

The system as recited in claim 1, wherein the business system is a Cross Enterprise business system (Taylor, pages 296 – 299, wrappers).

**Claim 7**

The system as recited in claim 1, wherein the plurality of heterogeneous components of the business system includes one or more legacy mainframe systems (Taylor, pages 296 – 299, wrappers).

**Claim 11**

The system as recited in claim 1, wherein the design patterns include one or more Mainframe integration and interoperability design patterns (Taylor, pages 296 – 299, wrappers).

**Claim 12**

The system as recited in claim 11, wherein the Mainframe integration and interoperability design patterns include one of a Synchronous Mainframe Web Services design pattern (web, page 56 and 63 – RPC) and an Asynchronous Mainframe Web Services design pattern (Web, page 273-279 – ability to process a method).

6. Claims 15 and 16 rejected under 35 U.S.C. 103(a) as being unpatentable over Web as applied to claim 1 above, and further in view of the Hazards of Closed-Process Development, Mark Smith, February 2003.

**Claim 15**

The system as recited in claim 1, wherein the design patterns include one of a Closed Process integration design pattern and an Open Process integration design pattern (Closed, page 1). Web teaches implementation of Web integrated services but does not explicitly mention Closed Process. It is Closed who teaches Closed-Process. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to implement a closed process design pattern.

Art Unit: 2193

**Claim 16**

The system as recited in claim 15, wherein the design patterns include one of a Service Consolidation-Broker integration design pattern and a Reverse Auction-Broker integration design pattern.

**Interpretation**

Present limitations are deemed data. The ability to select from lists is supported by presentation layer of Web. the content is data.

7. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Web as applied to claim 1 above, and further in view of Essential Guide to Object Monitors, Karen Boucher et al, March 1999 (Monitor).

**Claim 17**

The system as recited in claim 1, wherein the layers comprise two or more of: a network layer configured to serve as an underlying network for integrated Web Services implemented according to the integrated Web Service architecture ; a transport layer for delivering messages between components of the integrated Web Services (Web, pages 84-85, MIME or page 13, TCP); a service description language layer configured to describe service type (See claim 1 WSDL) and functionality of the integrated Web Services (See claim 1 – designing the application); a transaction routing layer configured to route messages on the transport layer (TCP above and Web, page 145) ; a service discovery layer configured to search for (HTTP below) and locate the integrated Web Services; a service negotiation layer configured to negotiate exchanges between service requesters and service providers implemented according to the integrated Web Service architecture (Web, page 273-279); a management layer configured for provisioning of the integrated Web Services (Web, page 52-55) and for monitoring (Monitor, page pages 45- 47) and administration of the integrated Web Services (Web, page 221); a Quality of Service layer configured to provide reliability (Web, pages 99, 127), scalability (Web, inherent with object technology) , and availability for the integrated Web Services (see claim 1) ; a security layer configured to provide authentication (Web, page 14) , entitlement (Web, page 87), and nonrepudiation security on the transport layer (Web, page 13); and an Open Standards layer (Web, page 13). Web teaches a flexible integrated environment for web services but does not explicitly teach monitoring. It is Monitor who teaches monitoring (Monitor, pages 45-47). Therefore, it would have been obvious to one of ordinary skill in the art to combine the teachings of Web and Monitor because monitors make systems more reliable.

***Response to Arguments***

8. Applicant's arguments filed February 26, 2008 have been fully considered but they are not persuasive.

**Applicant's Argument**

“Section 102(e) Rejection:

The Examiner rejected claims 1-4, 8-10 and 13-14 under 35 U.S.C. § 102(e) as being anticipated by Conallen ("Building Web Applications with UML", Second Edition). Applicant respectfully traverses this rejection for at least the following reasons.

In regard to claim 1, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications. On page 3, first paragraph, Conallen states (emphasis added), "this book is about building model-driven Web applications." The term Web Services is well known in the art, and one of ordinary skill in the art would recognize the difference between the terms Web Services and Web applications. The background section of the instant application provides an extensive discussion of Web Services. Furthermore, Conallen defines Web applications thusly in the paragraph beginning on page 9 and extending onto page 10 (emphasis added):

In its simplest terms, a Web application is a Web system that allows its users to execute business logic with a Web browser...There is a subtle difference between a Web application and a Web site. For the purpose of this book, a Web application is a Web site where user input - navigation through the site and data entry - affects the state of the business: beyond, of course, access logs and hit counters. In essence, a Web application uses a Web site as the front end to a business application.

Conallen does briefly discuss Web Services on pp. 63-68, in a section titled "Web Services" that appears in Chapter 4, titled "Beyond HTTP and HTML," which begins on page 49. Conallen makes clear the distinction between Web Services and Web applications, for example in the third paragraph on page 63:



The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, it is not a Web application- specific technology. Instead, it uses Web technologies, such as Web servers and HTTP, to provide a set of services that can be invoked by other programs on the network.

On page 49, second paragraph, Conallen discusses the content and reason for Chapter 4:

With the recent successes of Web applications, more and more architects are choosing this architecture for their next generation of systems. The significant advantages of easy deployment and minimal client configuration are well suited to organizations that maintain a varied array of computer types and models. This increased use of the Web as an architectural platform, however, has stretched the limits of the ability for HTTP and HTML to deliver the functionality required in relatively sophisticated software systems. This chapter discusses the limitations and extensions to these two principal elements of Web applications: HTTP and HTML.

Thus, both Conallen and Applicant's specification are consistent in distinguishing Web Services from Web applications. The term "Web Services" is a well-understood term of art. Most of the teachings of Conallen cited by the Examiner pertain to Web applications, not Web Services as recited in Applicant's claim. Nowhere does Conallen extend the notions presented in his book to generating integrated Web Service architectures. Again, Conallen clearly states "this book is about building model-driven Web applications." Conallen's discussion of Web Services is simply an aside "discuss[ing] the limitations and extensions to...two principal elements of Web applications."

Thus, contrary to the Examiner's assertion, Conallen does not anticipate a system for integrating Web Services with a business system. The Examiner cites Conallen, page 9, last paragraph, in support of this assertion. However, this citation simply states (emphasis added):

A Web application builds on and extends a Web system to add business functionality.

Again, as noted in the background section of the instant application and in the Conallen reference itself, Web applications are clearly and distinctly different than Web Services

In further regard to claim 1, Conallen does not anticipate program instructions executable by the processor to: generate an integrated Web Service architecture. The Examiner first cites Conallen, page 10, Figure 2-1. Figure 2-1 simply shows a block diagram of a "basic Web system." Page 10's text simply includes a portion of a paragraph, cited above, that defines Web applications and thus distinguishes between the notion of Web Services and Web applications, and an introduction to a discussion of HTTP. These citations clearly do not teach the limitations as recited in the claim. These citations do not teach "program instructions executable by the processor to" do anything, much less program instructions executable by a processor to generate an integrated Web Service architecture. Nowhere does Conallen teach a system for integrating Web Services with a business system, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to: generate an integrated Web Service architecture. Conallen's book does not teach any such system comprising a processor and memory comprising program instructions to perform the elements as actually recited in claim 1 when considered as a whole. In fact, while Conallen's book is "about building model-driven Web applications," Conallen does not even teach a system comprising: a

processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to: generate a Web application architecture.

The Examiner then cites pp. 65-66, 64-65, and 66-68 along with page 10, Figure 2-1 as teaching program instructions executable by the processor to: generate an integrated Web Service architecture. As noted above, these citations are from Conallen's discussion of Web Services that is simply an aside "discuss[ing] the limitations and extensions to...two principal elements of Web applications." These citations simply describe UDDI, SOAP and WSDL, respectively. However, like the above citations, these citations do not teach "program instructions executable by the processor to" do anything, much less program instructions executable by a processor to generate an integrated Web Service architecture. Nor do the two citations in combination teach the limitations as recited in claim 1.

In further regard to claim 1, Conallen does not anticipate program instructions executable by the processor to: generate an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system in accordance with one or more integration design patterns. The Examiner cites Conallen, page 425, "Master Template Pattern or classes - one example in Figure 6-11, page 115." Page 425 appears in Appendix D, titled "Master Template Pattern," which begins on page 423. The first paragraph on page 423, titled "Overview," states (emphasis added):

The master template mechanism was influenced by the Java Pet Store 1.0.1 example documented in the Java BluePrints. In this mechanism, one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel and providing a single

source for updates. This mechanism is most useful for applications that can benefit from an explicitly controlled user interface template.

Page 425 includes further discussion of "screen templates" and "class diagrams." Appendix D, which discusses a "master template mechanism" in which "one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel" clearly does not describe anything at all like one or more integration design patterns used in generating an integrated Web Service architecture comprising a plurality of heterogeneous components" of the business system. Furthermore, Figure 6-11 on page 115 illustrates a "requirements set", discussion of which starts on page 114. This Figure and section has nothing whatsoever to do with Appendix D, and, contrary to the Examiner's assertion, is nowhere in Conallen described as "one example" of a Master Template Pattern. The two citations appear to have little if any relation with one another.

In further regard to claim 1, Conallen does not anticipate wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate one or more Use Cases for the integrated Web Service. The Examiner cites Conallen, pp. 173-185, 120, 216, 411-414, 101-105, 177-179, 139-141, and 179-183. While the Conallen reference includes several sections that discuss various "use cases," nowhere does Conallen disclose one or more Use Cases for an integrated Web Service. As previously noted, Conallen is generally directed at building Web applications ("this book is about building model-driven Web applications"), not Web Services. The citations provided by the Examiner disclose various Web application use cases. On page

173, one of the pages cited by the Examiner, at the beginning of a section titled "Use Cases", Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications.

Furthermore, while Conallen does disclose various Use Cases (again, Web application-specific use cases), nowhere does Conallen disclose program instructions executable by a processor to generate one or more Use Cases for an integrated Web Service. In fact, Conallen does not even disclose program instructions executable by a processor to generate one or more Use Cases in reference to the Web application-specific Use Cases Conallen does discuss.

In further regard to claim 1, Conallen does not anticipate wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service. The Examiner asserts "as per above" in reference to this limitation. Applicant's above arguments make it clear that Conallen is not even directed at generating architectures for integrated Web Services. Furthermore, Conallen does not even disclose program instructions executable by a processor to generate a high-level architecture of any type. Conallen nowhere discloses any such computer-executable program instructions.

In further regard to claim 1, Conallen does not anticipate wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the

Art Unit: 2193

integrated Web Service. The Examiner cites Conallen, page 438. This citation is from Appendix E, "Glossary Application," that begins on page 429. The second paragraph on page 429 states (emphasis added):

The overall goal and vision for this application is to demonstrate, in the context of a simple and functional application, a technique for modeling Web applications with UML.

This Appendix is clearly directed at a technique for modeling Web applications, and an example thereof, and not at Web Services, and thus none of the examples from this Appendix, including page 438, illustrate anything like a high-level architecture[for an integrated Web Service] that identifies two or more entities of the integrated Web Service.

In further regard to claim 1, Conallen does not anticipate wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities. The Examiner cites page 177, "relationship." This citation appears in a section titled "The Use Case Model" that begins on page 176, and specifically discusses relationships between "actors" and Use Cases (page 177, first paragraph) and relationships between Use Cases (page 177, second paragraph). This citation clearly describes nothing like what is actually recited in claim 1 (the relationship among two or more entities of an integrated Web Service).

In further regard to claim 1, Conallen does not anticipate wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to generate a logical architecture for the integrated Web Service according to the Use Cases, wherein the

logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers. The Examiner cites Conallen, pages 237-242. This citation is in a section titled "Web

Application Extension [WAE] for UML" that begins on page 236. Again, this section is directed at Web applications, not Web Services. The subsection beginning on page 237, cited by the Examiner, is titled "Logical View." This subsection, like the rest of the section, is clearly directed at Web applications, not Web Services.

Furthermore, Conallen, in this section or elsewhere, does not even disclose program instructions executable by a processor to generate a logical architecture of any type.

Applicant reminds the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 221 USPQ 481,485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. Richardson v. Suzuki Motor Co., 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As shown by the Applicant's above arguments, nowhere does the Conallen reference disclose "each and every element of the claimed invention" (claim 1 of the instant application) as arranged in the claim. Furthermore, even if Conallen did disclose one or more of the above elements, nowhere does Conallen disclose the above elements arranged as in claim 1. Moreover, the Examiner has improperly cited portions of Conallen from various chapters, sections, and appendices, some of which do not appear to be directly related, in an attempt to

assert that Conallen discloses what is recited in claim 1. For at least the reasons given above, Conallen clearly does not anticipate Applicant's claim 1.

**Section 103(a) Rejection:**

The Examiner rejected claims 5-7 and 11-12 under 35 U.S.C. § 103(a) as being unpatentable over Web in view of Taylor ("Object-Oriented Information Systems Planning and Implementation"). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time

The Examiner rejected claims 15 and 16 under 35 U.S.C. § 103(a) as being unpatentable over Web in view of Smith ("The Hazards of Closed-Process Development"). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time.

The Examiner rejected claim 17 as being unpatentable over Web as applied to claim 1 above, and further in view of Boucher, et al. ("Essential Guide to Object Monitors"). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time.

Applicant also asserts that the rejection of numerous ones of the dependent claims is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time."

**Examiner's Response**



Applicant's argument hinges on the idea that just because the Computer Aided Software Engineering (CASE) tool says it is for building Web Applications it can not be used to build Web Services. Examiner disagrees, the reference not only provides a CASE tool but also specifically discloses Web Service Description Language. WSDL is intended to make development of Web Services easier. Prior to using UML for building Web Applicants it was commonly used for non web software projects. Software Engineers did not refuse to use it because it does not say Web on it. The reference provides an Industry Groups modeling technique (UML) in a case tools and provides a programming language specifically for building Web Services.

Applicant's argument that the examples are not Web Services is not persuasive. The results of the use of the CASE tool is one must model the problem domain in order to generate the solution. In object oriented technology this typically results in classes that represent the real world. Specifically the claim limitations that exclude using the disclosed tool to model and generate a Web Service using WSDL is not present.

Applicant's argument is not persuasive.

### ***Conclusion***

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

### ***Correspondence Information***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Todd Ingberg/  
Primary Examiner